

OULU UNIVERSITY OF  
APPLIED SCIENCES



Balakumar Chittu Gnanamoorthy

**BEST SALE MOBILE APPLICATION DEVELOPMENT**

# **BEST SALE MOBILE APPLICATION DEVELOPMENT**

Balakumar Chittu Gnanamoorthy  
Master's Thesis  
Spring 2014  
Master of Information Technology  
Oulu University of Applied Sciences

## **ABSTRACT**

Oulu University of Applied Sciences  
Degree Programme in Information Technology

---

Author: Balakumar Chittu Gnanamoorthy  
Title of thesis: Best Sale Mobile Application  
Supervisor: Kari Laitinen  
Term and year of completion: Spring 2014

Number of pages: 43

---

The aim of this thesis was to find a best way to advertise short-term sales and reduce paper waste. Another aim was to help the seller to identify a potential customer for a specific product, by means of providing a number of people accessing an advertisement.

A mobile phone is one of the widely used electronic devices by many people. Modern mobile phones support a good internet connection. Also, people carry their mobile devices with them even outside their homes. So a mobile application would be a best way to advertise short-term sales. The benefits include reducing the time to reach advertisements from a seller to a buyer, reducing paper waste, and saving money.

A client mobile application “Best Sale Application” was implemented to show an advertisement to a user and a back end server was developed to feed the advertisements and provide them to a client application on demand. The developing of a prototype version of Best Sale Application is explained in this thesis. The application runs on Windows Phone and it is able to interact with an early version of Best Sale Server. The back end server side development is explained in the Thesis paper of Ramkumar Patturaj entitled “Server for Best Sale Application”.

The proto type version of Best Sale Application carries a significant potential for a future upgrade to a real product to be launched in the Windows Market place for a real user. It will be very useful for a shop owner to reach a potential customer faster and save money compared to a paper advertisement.

---

Keywords: Windows Phone Application, Sale, Advertisement.

## **PREFACE**

This thesis represents a prototype version of a mobile application for advertising short-term sales.

Firstly, I would like to express my sincere thanks to Oulu University of Applied Sciences for providing a Master Degree Programme which is suitable for full-time working people like me. Then to Ramkumar Patturaj, my best friend and thesis mate who helped me to shape my best sale ideas and supported me to develop the backend server.

My sincere thanks goes to Dr Kari Laitinen for his humble, yet truly admirable ability to help writing this work in better words.

Last, but by no means least, I thank my parents and my family, specially my wife Rubini Matta Santharam and son Shashvath Chittu Balakumar for their constant support and encouragement to complete my master degree programme.

Oulu, Finland, May 2014

**Balakumar Chittu Gnanamoorthy**

## TERMS AND ABBREVIATIONS

|          |   |
|----------|---|
| RSS      | - Rich Site Summary                       |
| URL      | - Uniform Resource Locator                |
| UI       | - User Interface                          |
| HTML     | - Hyper Text Mark-up Language             |
| XAML     | - Extensible Application Mark-up Language |
| OS       | - Operating System                        |
| SD Card  | - Secure Digital Card                     |
| IP       | - Internet Protocol                       |
| API      | - Application Program Interface           |
| HTTP     | - Hyper Text Transfer Protocol            |
| REST API | - Representational State Transfer         |
| OPML     | - Outline Processor Mark-up Language      |

## LIST OF FIGURES

|   |    |
|---|----|
| Figure 1. Windows Phone Application Platform Architecture.....          | 14 |
| Figure 2. Windows Phone Application Bar .....                           | 14 |
| Figure 3. Windows Phone Supported Themes.....                           | 16 |
| Figure 4. Windows Phone Application Frame Structure.....                | 17 |
| Figure 5. Use Case diagram of Best Sale Mobile Application.....         | 22 |
| Figure 6. Activity Diagram of Best Sale Mobile Application.....         | 24 |
| Figure 7. Best Sale Server and Best Sale Application Communication..... | 27 |
| Figure 8. Best Sale Server and Application Sequence diagram.....        | 28 |
| Figure 9. Best Sale Application in the Windows Phone Menu.....          | 31 |
| Figure 10. Best Sale Application Splash Screen Page.....                | 31 |
| Figure 11. Best Sale Application Main Panorama Page.....                | 32 |
| Figure 12. Beset Sale Application Store Product View.....               | 33 |

## LIST OF TABLES

|  |    |
|--|----|
| Table 1. Windows Phone Installation Folder.....            | 18 |
| Table 2. Windows Phone Local Folder.....                   | 19 |
| Table 3. Windows Phone External Storage.....               | 20 |
| Table 4. Application Launch Test Case.....                 | 34 |
| Table 5. Application City List Verification Test Case..... | 34 |
| Table 6. Shop Name Verification Test Case.....             | 34 |
| Table 7. Product List Verification Test Case.....          | 34 |
| Table 8. Product Details Verification Test Case.....       | 35 |
| Table 9. Removed Product Verification Test Case.....       | 35 |
| Table 10. Android and iOS differences.....                 | 39 |



## TABLE OF CONTENTS

|   |    |
|---|----|
| ABSTRACT.....                                   | 3  |
| PREFACE.....                                    | 5  |
| TERMS AND ABBREVIATIONS.....                    | 6  |
| LIST OF FIGURES.....                            | 7  |
| LIST OF TABLES.....                             | 8  |
| CONTENTS.....                                   | 9  |
| 1 INTRODUCTION.....                             | 11 |
| 1.1 Motivation and background.....              | 11 |
| 1.2 Objectives.....                             | 12 |
| 2 TECHNOLOGIES.....                             | 13 |
| 2.1 About Windows Phone OS.....                 | 13 |
| 2.2 Windows Phone OS Architecture .....         | 13 |
| 2.3 App bars.....                               | 14 |
| 2.4 Themes.....                                 | 15 |
| 2.5 In-App navigation.....                      | 17 |
| 2.6 Data storage.....                           | 18 |
| 2.6.1 Installation folder.....                  | 18 |
| 2.6.2 Local folder.....                         | 19 |
| 2.6.3 External folder.....                      | 20 |
| 3. DESIGN AND IMPLEMENTATION .....              | 21 |
| 3.1 Requirement specifications.....             | 21 |
| 3.2 Analysis and Design of the Application..... | 23 |
| 3.3 Implementing the Best Sale Application..... | 25 |

|   |    |
|---|----|
| 3.3.1 Opml.xml file and its purpose.....        | 25 |
| 3.3.2 RSS model.....                            | 26 |
| 3.3.3 User Interface.....                       | 30 |
| 3.3.4 Testing.....                              | 33 |
| 4. FUTURE R&D POSIBILITIEIS.....                | 36 |
| 4.1 Extending Best Sale App functionality.....  | 36 |
| 4.2 Implementation for PC, Andriod and iOS..... | 38 |
| 4.3 Effort Estimations.....                     | 41 |
| 5. CONCLUSIONS .....                            | 42 |
| 6. REFERENCES .....                             | 43 |

# 1. INTRODUCTION

In traditional advertising an advertisement is delivered to homes of potential buyers. Often buyers do not read all advertisements which come from different shops. Other reason could be that buyers do not have enough time to read them at home. Instead, a buyer may prefer to read while he/she is travelling in a bus or train or during his/her coffee hours at office. Carrying many advertisements just to read them when the buyer has spare time is not a feasible solution. But if we convert advertisement papers into an electronic format and push them into electronic devices like mobile phones, PCs and tablets, where buyer will have access to any one of them when he/she has spare time to read it, will definitely be profitable to the seller.

## 1.1 Motivation and background

In Finland many stores have a short-term sale period. They advertise in traditional newspapers. These newspapers are delivered by post to homes of people who are staying in the same city. The sale duration varies from three days to one week depending on the sale type. Such sale advertisement papers usually arrive to homes by post a few days before the sale begins.

Often I started throwing those sale advertisement papers away without reading them. I saw the same habit among my neighbours. There could be many reasons why this happens, for example, a home where only adults are living a sales advertisement on kid's dress and accessories is delivered. Other reason could be that there are people who spend more time in travelling to work and back home and they prefer to read such advertisement during the travelling time.

This triggered a few questions in my mind. Why is a seller sending an advertisement paper to a home where no one is interested in it? How a seller could get the information whether those advertisement papers are read by someone or not?

Should there be a better way to provide such information so that anyone can access it anytime from anywhere? How could we avoid such huge waste of papers?

Answers to these questions gave me a motivation to develop a mobile application that could list all short-term sale advertisements. The user of this application can read an advertisement from anywhere using his/her mobile device. A potential background server could create a statistic for the seller to understand how many users have read his/her store advertisement.

## **1.2 OBJECTIVES**

The mobile application that was developed as answer to the above questions is named Best Sale. The target audience for Best Sale mobile application is all buyers who are interested in buying items when they go on sale for a specific duration in shops, especially seasonal offers like Christmas sale, summer sale. The first version of Best Sale mobile application runs on a Windows Phone which has Windows Phone OS 7 and 8. A backend server has all the latest short-term advertisements ordered by city and by shop in its own database. This mobile application retrieves them from the backend server and shows them in the user's mobile in a simplified readable format with rich UI elements.

## **2. TECHNOLOGIES**

In Finland Nokia has played a major role on mobile handset market. A few years ago Nokia started making products using Microsoft Windows Phone OS and recently Nokia mobile phones division was sold to Microsoft. The primary target of Best sale application is to be launched in Finland. Considering Nokia and its handset future models the first version of Best Sale Application is developed using Microsoft Windows Phone OS 7.5. Also, Windows Phone application developed using Windows Phone OS 7.5 runs on Windows Phone OS 8. But the other way is not supported.

### **2.1 ABOUT WINDOWS PHONE OS**

The first version of Windows Phone 7 was released in 2010. For the user it was unique compared to the other mobile OS on the market. The live tiles provided an efficient way to update apps data in the Windows home screen. The developer platform offered powerful tools for a rapid application development. The Windows 8 OS offers more opportunities for developers. An application developed using Windows 8 tools can run on a mobile, PC and Windows 8 tablet.

### **2.2 Windows Phone OS Architecture**

The Windows Phone applications are controlled by three navigation buttons: Back, Start and Search. Windows Phone runtime includes Silver light and XNA Framework which provides secure and rich UI applications. Visual Studio, Expression Blend helps the developers to create, debug, deploy and update the applications [1].

Windows Market place provides the tools and supports to the developers to release their application for Windows Phones. The Figure 1 shows the Windows Phone OS application platform architecture diagram [1].

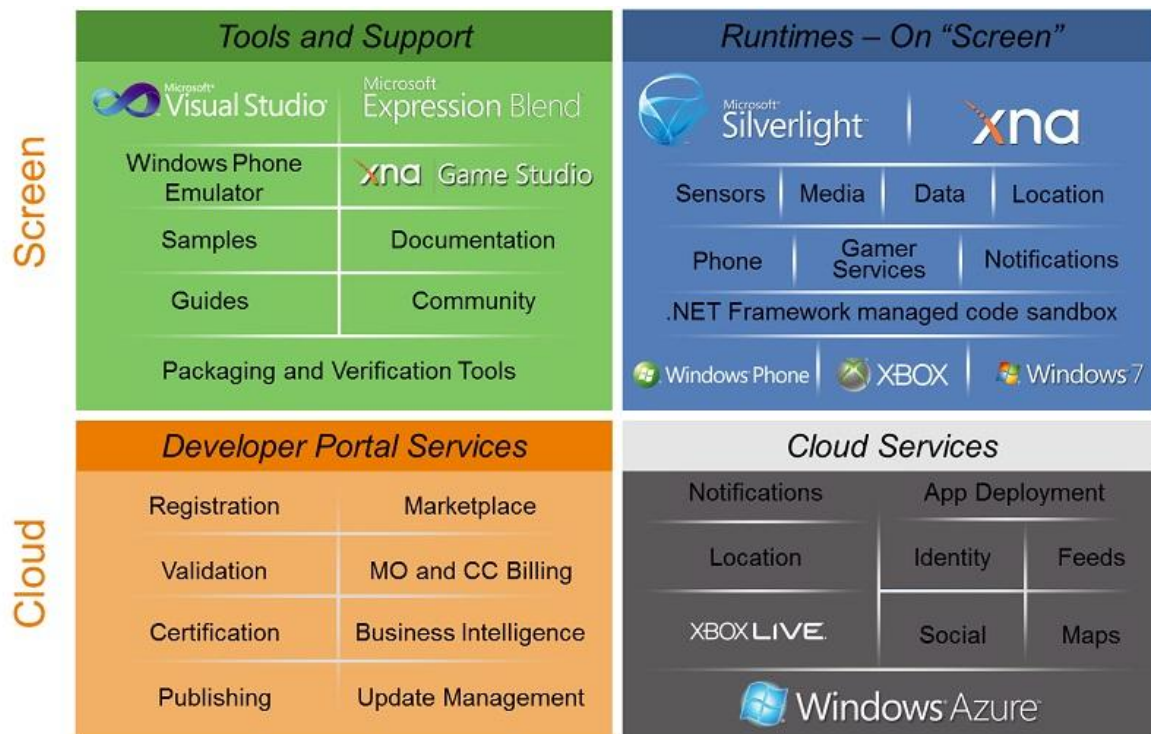


FIGURE 1. Windows Phone Application Platform Architecture [1]

## 2.3 App bar

The Application Bar has icon buttons and labels. The labels are by default hidden. A User can click the ellipsis to display the labels. By default the application bar will adopt the phone orientation.

Figure 2 illustration shows an example of an Application Bar that uses the default size.



FIGURE 2. Windows Phone Application Bar [1]

## 2.4 Themes

The application theme is defined using resource and resource dictionaries. The resources are loaded and applied when an application starts. It is recommended to use build-in themes for the applications components which will be consistent with the device theme. The Figure 3 shows the supported theme for Windows Phone 7.5 and Windows Phone 8 OS version.

| ACCENT<br>COLOR  | RED, GREEN, BLUE                              | HEX                                   | WINDOWS<br>PHONE 7.5 | WINDOWS<br>PHONE 8 |
|------------------|---|---------------------------------------|----------------------|--------------------|
| Lime             | 164, 196, 0<br><small>162, 193, 57</small>    | #FFA4C400<br><small>#FFA2C139</small> | ✓                    | ✓                  |
| Green            | 96, 169, 23<br><small>51, 153, 51</small>     | #FF60A917<br><small>#FF339933</small> | ✓                    | ✓                  |
| Emerald          | 0, 138, 0                                     | #FF008A00                             |                      | ✓                  |
| Teal<br>Viridian | 0, 171, 169                                   | #FF00ABA9                             | ✓                    | ✓                  |
| Cyan<br>Blue     | 27, 161, 226                                  | #FF1BA1E2                             | ✓                    | ✓                  |
| Cobalt           | 0, 80, 239                                    | #FF0050EF                             |                      | ✓                  |
| Indigo           | 106, 0, 255                                   | #FF6A00FF                             |                      | ✓                  |
| Violet<br>Purple | 170, 0, 255                                   | #FFAA00FF                             | ✓                    | ✓                  |
| Pink             | 244, 141, 208<br><small>230, 113, 184</small> | #FFF472D0<br><small>#FFE671B8</small> | ✓                    | ✓                  |
| Magenta          | 216, 0, 115                                   | #FFD80073                             | ✓                    | ✓                  |
| Crimson          | 162, 0, 37                                    | #FFA20025                             |                      | ✓                  |
| Red              | 229, 20, 0                                    | #FFE51400                             | ✓                    | ✓                  |
| Orange<br>Mango  | 250, 104, 0<br><small>240, 150, 9</small>     | #FFFA6800<br><small>#FFF09609</small> | ✓                    | ✓                  |
| Amber            | 240, 163, 10                                  | #FFF0A30A                             |                      | ✓                  |
| Yellow           | 227, 200, 0                                   | #FFE3C800                             |                      | ✓                  |
| Brown            | 130, 90, 44<br><small>160, 80, 0</small>      | #FF825A2C<br><small>#FFA05000</small> | ✓                    | ✓                  |
| Olive            | 109, 135, 100                                 | #FF6D8764                             |                      | ✓                  |
| Steel            | 100, 118, 135                                 | #FF647687                             |                      | ✓                  |
| Mauve            | 118, 96, 138                                  | #FF76608A                             |                      | ✓                  |
| Sienna           | 160, 82, 45                                   | #FFA0522D                             |                      | ✓                  |

BLUE TEXT denotes Windows Phone 7.5 names and values

FIGURE 3. Windows Phone Supported Themes [1]



## 2.5 In-App Navigation

A windows application can navigate forward through a screen content but it is advised to use the back button for backward. Figure 4 shows the Windows Phone Application frame structure. The advantages of such model are

- Easily created view-based applications that fit naturally into the Windows Phone navigation model [1]
- Provided default transitions that match the Windows Phone look and feel [1]

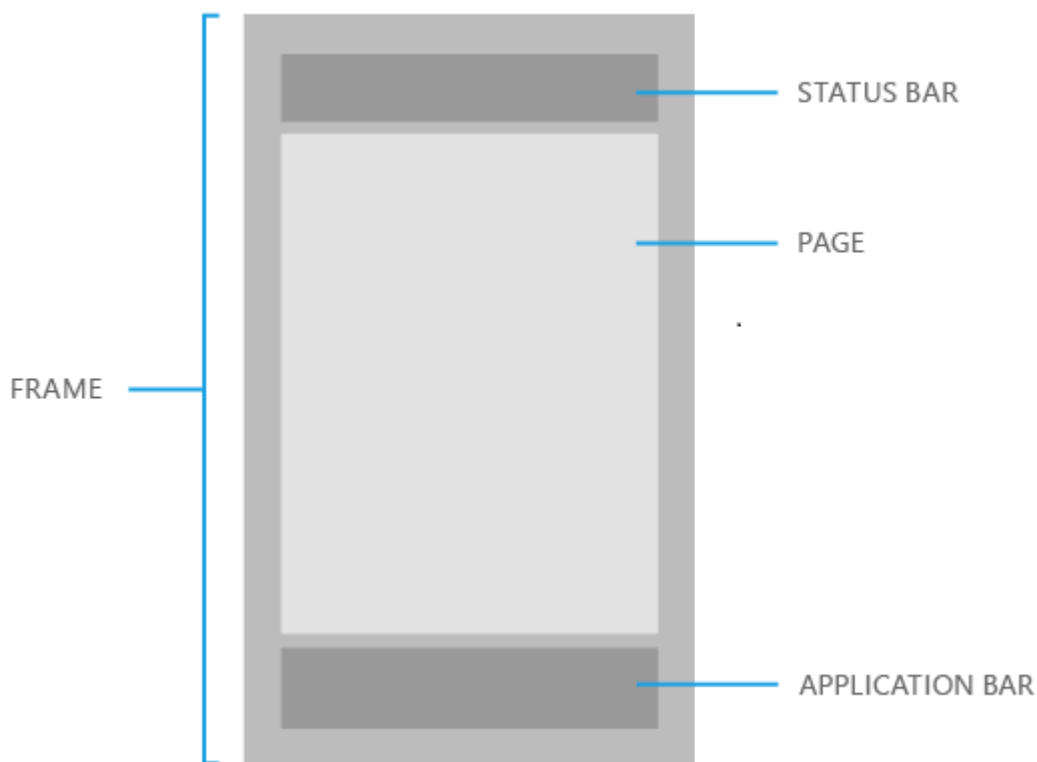


FIGURE 4. Windows Phone Application Frame Structure [1]

**Frames:** Each application has one master frame the look and feel of which integrated into windows Phone.

**Pages:** Application contents are loaded in the page area.

**Status Bar:** Mainly used to update the system-level status information and other important application notification.

**Application Bar:** Provides a shortcut within the application for common tasks. It can be extended to add menu items.

## 2.6 Data Storage

### 2.6.1 Installation Folder

Installation folder for the application is a read-only folder which can be accessed using APIs shown in Table 1.

TABLE 1. Windows Phone Installation Folder [1]

| API   | Description   | Supported in Windows Phone OS 7.1 | Supported in Windows Phone 8 |
|---|---|-----------------------------------|------------------------------|
| <a href="#">GetResourceStream</a>                             | Use this method to stream files from the installation folder. For an example, see <a href="#">How to deploy a reference database with an app for Windows Phone</a> .  | ✓                                 | ✓                            |
| <a href="#">DataContext</a>                                   | To connect to a reference database that is part of the app package, use the installation folder protocol in the path when you create your local database object. For examples, see <a href="#">Local database connection strings for Windows Phone</a> .      | ✓                                 | ✓                            |
| <a href="#">InstalledLocation</a>                             | The <a href="#">InstalledLocation</a> property of the current <a href="#">Package</a> class returns the installation folder as a <a href="#">StorageFolder</a> object. Use the <a href="#">Path</a> property to get the full path of the installation folder. | ✗                                 | ✓                            |
| <a href="#">StorageFile</a> and <a href="#">StorageFolder</a> | Use these APIs to work with files and folders in the installation folder.   | ✗                                 | ✓                            |

## 2.6.2 Local Folder

Each Windows Application has its own isolated data storage folder and it is called a local folder. A Developer should use this to store application persistent data.

Table 2 shows the APIs used to access the local folder.

TABLE 2. Windows Phone Local Folder [1]

| API   | Description   | Supported in Windows Phone OS 7.1 | Supported in Windows Phone 8 |
|---|---|-----------------------------------|------------------------------|
| <a href="#">DataContext</a>                                   | Local database files are stored in the local folder. To connect to a local database, use the local folder protocol in the path when you create your local database object. For more info, see <a href="#">Local database for Windows Phone</a> .  | ✓                                 | ✓                            |
| <a href="#">IsolatedStorageSettings</a>                       | Work with key/value pairs in the local folder. For an example see, <a href="#">How to create a settings page for Windows Phone</a> .  | ✓                                 | ✓                            |
| <a href="#">IsolatedStorageFile</a>                           | Work with files and folders in the local folder. For an example, see <a href="#">Quickstart: Working with files and folders in Windows Phone 7</a> .  | ✓                                 | ✓                            |
| <a href="#">StorageFile</a> and <a href="#">StorageFolder</a> | Use these APIs to work with files and folders in the local folder. For an example of <a href="#">StorageFolder</a> , see <a href="#">Quickstart: Working with files and folders in Windows Phone 8</a> .  | ✗                                 | ✓                            |
| <a href="#">LocalFolder</a>                                   | The <a href="#">LocalFolder</a> property of the current <a href="#">ApplicationData</a> class returns the local folder as a <a href="#">StorageFolder</a> object. Use the <a href="#">Path</a> property to get the full path of the local folder. | ✗                                 | ✓                            |

### 2.6.3 External Storage

An application can use the APIs shown in Table 3 to access the SD card. But it is limited to only for those files which the application is registered to handle and it is read-only.

TABLE 3. Windows Phone External Storage [1]

| API   | Description   | Supported in Windows Phone OS 7.1 | Supported in Windows Phone 8 |
|---|---|-----------------------------------|------------------------------|
| <code>ExternalStorageDevice</code>  | Represents an SD card. Use the <code>ExternalStorageID</code> property to uniquely identify a card.                     | ✗                                 | ✓                            |
| <code>GetExternalStorageDevicesAsync</code>                               | Returns a collection of SD cards that are inserted in the phone. This collection will contain no more than one SD card. | ✗                                 | ✓                            |
| <code>ExternalStorageFolder</code>  | Represents a folder on the SD card. Use the <code>Path</code> property to access the folder later.                      | ✗                                 | ✓                            |
| <code>ExternalStorageFile</code>  | Represents a file on the SD card. Use the <code>Path</code> property to access the file later.                          | ✗                                 | ✓                            |
| <code>GetFileAsync(String)</code> and <code>GetFolderAsync(String)</code> | Obtain a file or folder directly from the external storage device object.   | ✗                                 | ✓                            |

### **3. DESIGN AND IMPLEMENTATION**

This chapter describes the design and implementation of the Best Sale Windows Phone application. Use cases and activities will be used to analyze the requirements and specifications, which will be followed by the actual implementation of the specified functionalities of the application.

#### **3.1 Requirement specifications**

The Best Sale Application is presented in an easy usable UI format to a user. When the application is launched from the Windows application menu, it should download all the store advertisements from the back end server. The UI should show the city names and shops that it supports to list the sale deals.

From the Main page of the application the user should be able to navigate to another city with a minimal user interaction. Once the user chooses a city the shops within that city should be listed with shop names and a logo if available. The user should be able to browse the sale items and choose one of the items to check for more details.

The use case diagram in Figure 5 shows what can be done with the application.

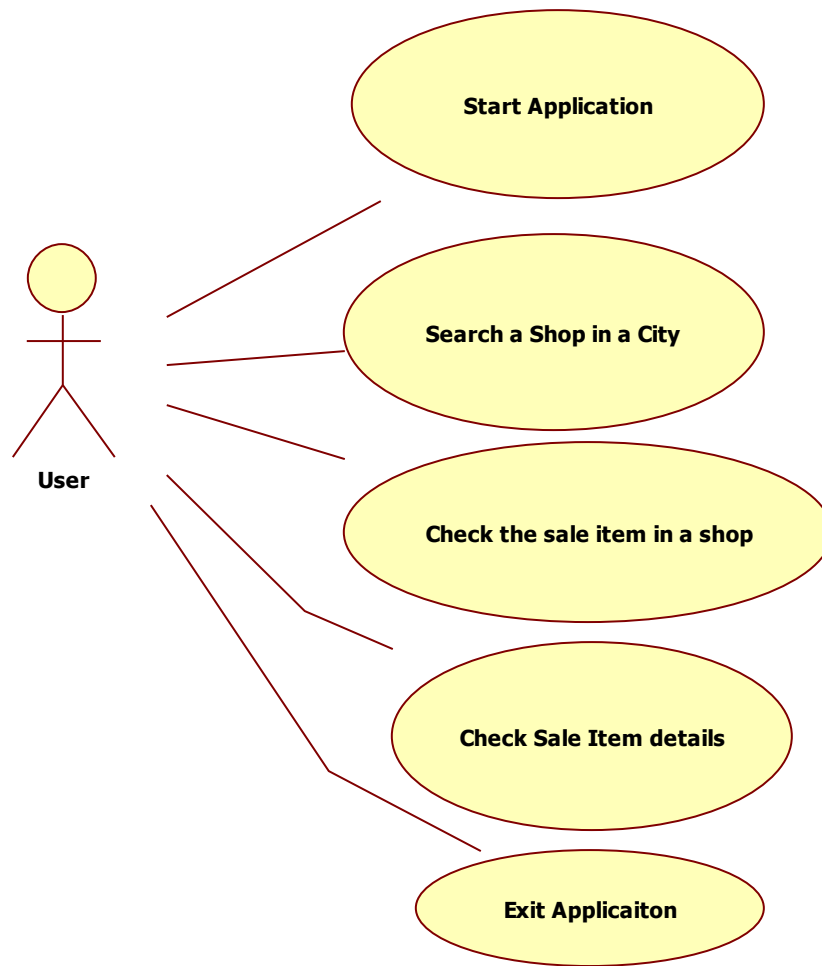


FIGURE 5. Use Case diagram of Best Sale Mobile Application

Start Application – A user selects the application “Best Sale” from the Windows phone menu. A splash screen with a Sale logo welcomes the user. On background at this time the application will download the OPML.xml file which has the store names and its RSS links. On first launch of the application it will download the first version of OPML.xml. If the application is launched for the second time, the application will download the OPML.xml only if there is a new version available, otherwise it will use the previously downloaded version. A new version of OPML is created in the server if new shops are added.

Search a Shop in a city – Launching the application will show a Windows panorama view as the first page of the application. Each view inside the panorama is called a panorama item. Each panorama item represents a city. The user can swipe left and right to see the other cities. Under one panorama item the user can swipe top and bottom to see the number of shops that have sale ongoing in that particular city.

Check the sale item in a shop- Once the user has selected a shop from a panorama item a new view is opened for that particular shop name. A list of images with three rows is displayed. The user can swipe right to see all sale items as images in that shop. Once the user reaches the end of the sale items, he/she can swipe left to search back.

Check Sale item details – From the shops view the user can tap on one of the images which represent a sale item. A new view is opened where the sale item image and product description is shown. It also shows the start and end date of sale for that particular product.

Exit application – In Windows phone an application can be closed only by pressing a back button in the device. To allow the application to be closed on every back press, the application returns to its previous view. When it reaches the main view, the back press will close the application.

### **3.2 Analysis and Design of the Application**

The application runs on Windows Phone 7.5 and later versions. To keep the application usage simple, the number of views it shows should be minimal and the navigation of the application pages should not be completed.



FIGURE 6. Activity Diagram of Best Sale Mobile Application.

The application is launched from the Windows menu, a welcome splash screen is shown for a few seconds for the first time. On background the application downloads the OPML file which has the RSS URL for shops. Once the splash screen disappears, a panorama view of the city names is shown. Under each city



name a list of shop names are shown. The user can swipe left or right to choose the city and then swipe up and down to choose the shop in a city. Once a shop has been chosen, a new view opens with a list of images of products. The user can swipe left first to see the product images and once the user has chosen a product, a new view with the product description and offer details is shown. The user can exit the application by pressing a back button in the Windows phone. To follow the Windows Phone application architecture every user back press brings the user to the previous view of the application until the user reaches the main view of the application and then exits the application.

### **3.3 Implementing the Best Sale Application**

This section describes the implementation of the actual application. The different resources and object model behind the application and user interface are explained in this section.

#### **3.3.1 Opml.xml file and its purpose**

The purpose of OPML (Outline Processor Mark-up Language) is to provide a way to exchange information between outliners and Internet services that can be browsed or controlled through an outliner. OPML is also a file format for an outline application, which is why OPML files may contain information about the size, position and expansion state of the window the outline is displayed in. OPML has also become popular as a format for exchanging subscription lists between feed readers and aggregators[2].

A <head> contains zero or more optional elements.

A <title> is the title of the document.

A <body> contains one or more <outline> elements.

An <outline> is an XML element containing at least one required attribute, text, and zero or more additional attributes. An <outline> may contain zero or more <outline> sub-elements. No attribute may be repeated within the same <outline> element.

The following OPML format is used in Best Sale Application.

```
<opml version="1.0">
  <head>
    <title>Finland Local Store Sale Deals</title>
  </head>
  <body>
    <outline title="Oulu" text="Oulu">
      <outline title="K-Market" type="RSS" xmlUrl="<RSS URL>"/>
      <outline title="S-Market" type="RSS" xmlUrl="<RSS URL>"/>
    </outline>
    <outline title="Jyväskylä" text="Jyväskylä">
      <outline title="Verkko-Kauppa" type="RSS" xmlUrl="<RSS URL>"/>
      <outline title="Prisma" type="RSS" xmlUrl="<RSS URL>"/>
    </outline>
    <outline title="Tampere" text="Tampere">
      <outline title="Karkainen" type="RSS" xmlUrl="<RSS URL>"/>
      <outline title="Sale" type="RSS" xmlUrl="<RSS URL>"/>
    </outline>
    <outline title="K-Market" type="RSS" xmlUrl="<RSS URL>"/>
    <outline title="S-Market" type="RSS" xmlUrl="<RSS URL>"/>
  </body>
</opml>
```

### 3.3.2 RSS model and Back end Server

Every shop should have its own RSS feed mapped in the back end server. Best sale application will download RSS and parse to retrieve the product images URL for each product. Then it will download those images from its corresponding URL.

Once an image has been downloaded, it gets cached in the device application storage area. So it is not necessary for the application to download the same image more than once. The downloaded images get deleted when the corresponding URL is removed from the backend server.

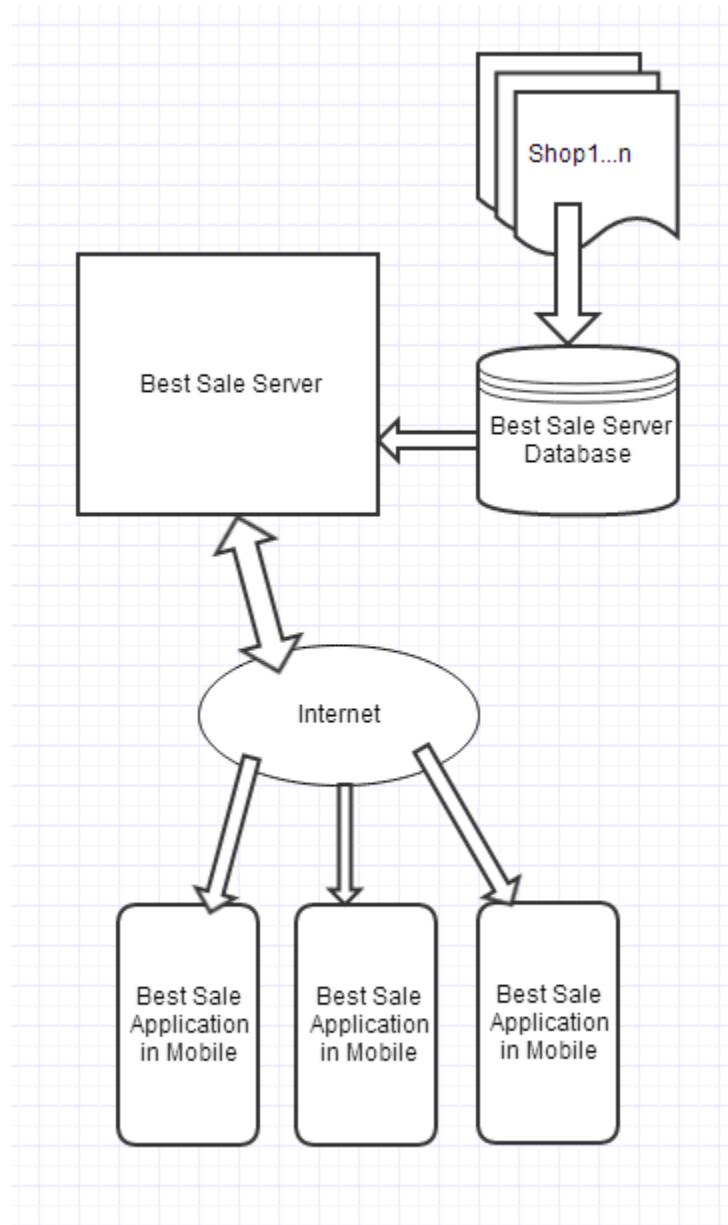


FIGURE 7. Best Sale Server and Best Sale Application Communication

The Best Sale Server has been developed as a separate project by another person. So explaining about the architecture of the back end server is out of the scope of this paper. As a simple definition the Best Sale Server provides the OPML file for the Best Sale Mobile Application which has the information about the shop and its URL.

The sequence diagram in Figure 8 shows the communication between the backend server and application

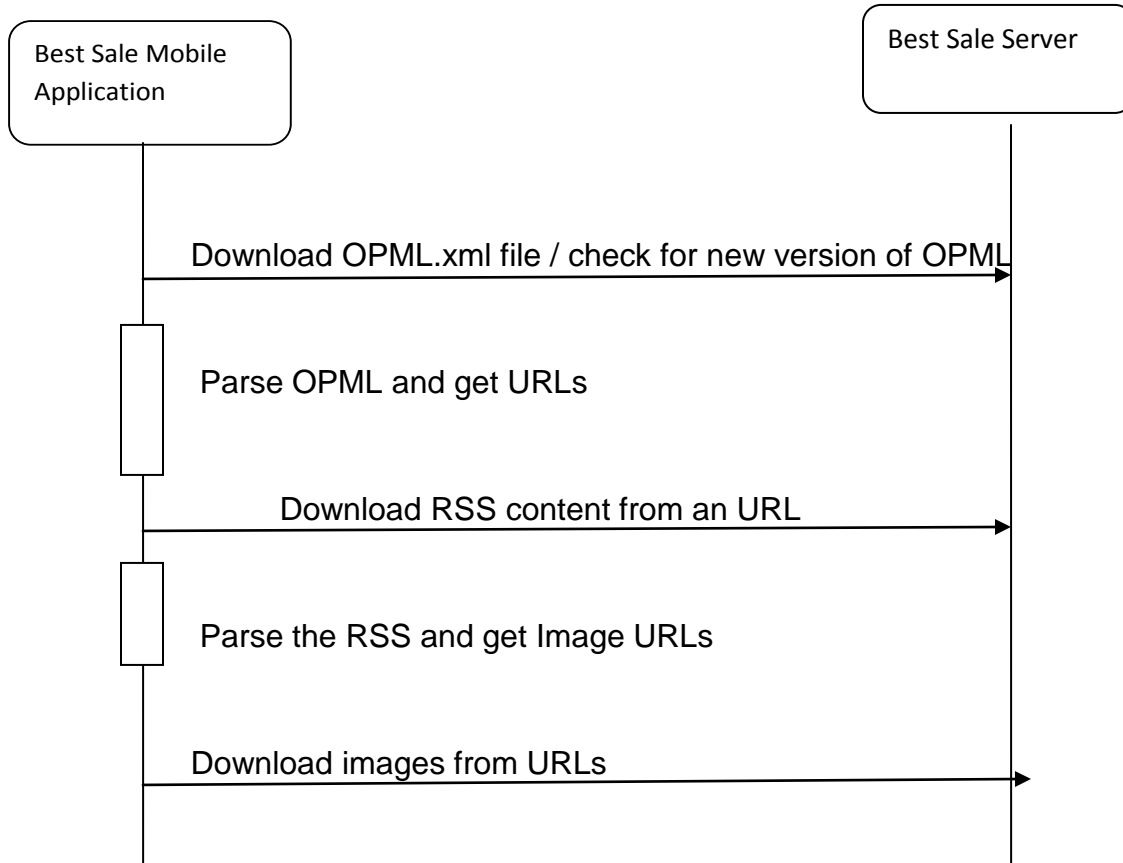


FIGURE 8. Best Sale Server and Application Sequence diagram

The following RSS model elements are used in the application and its functionalities:

- RSSPage object stores the URL of every shop in the application. It has a list of RSSFeed object to store the feed collection from the RSS URL.
- RSSFeed object stores the RSSItems from a feed.
- RSSItem object contains the actual product detail. It has a product image URL, item description, offer duration in dates.

- RSSService is the actual service object which downloads the content from the URL. It uses the Microsoft WebClient interface to download the content from the backend server.

A Code sample from the Best Sale Mobile application which is used for downloading an RSS content from a URL

```
public static void GetRSSFeedImageUri(RSSFeed feed, Action<Uri, RSSFeed>
onGetRSSFeedImageUriCompleted = null, Action<Exception> onError = null)
{
    WebClient webClient = new WebClient();

    webClient.OpenReadCompleted += delegate(object sender, OpenReadCompletedEventArgs e)
    {
        try
        {
            if (e.Error != null)
            {
                if (onError != null)
                {
                    onError(e.Error);
                }
                return;
            }
            XmlReader response = XmlReader.Create(e.Result);
            SyndicationFeed rssFeed = SyndicationFeed.Load(response);
            if (onGetRSSFeedImageUriCompleted != null)
            {
                onGetRSSFeedImageUriCompleted(rssFeed.ImageUrl, feed);
            }
        }
        catch (Exception error)
        {
            if (onError != null)
            {
                onError(error);
            }
        }
    };

    webClient.OpenReadAsync(new Uri(feed.URL));
}
```

RSSCache is the object which stores the downloaded product details in the application storage area. It uses the Microsoft IsolatesStoreageArea object to access the application storage.

The following code shows how the application reads data from the storage.

```
public static RSSCache Load()
```

```

{
    RSSCache cache = null;
    using (IsolatedStorageFile file = IsolatedStorageFile.GetUserStoreForApplication())
    {
        using (IsolatedStorageFileStream stream = new
IsolatedStorageFileStream("rssCache.dat", System.IO.FileMode.OpenOrCreate, file))
        {
            if (stream.Length > 0)
            {
                App.Log("Reading cache from file");
                DataContractSerializer serializer = new DataContractSerializer(typeof(RSSCache));
                cache = serializer.ReadObject(stream) as RSSCache;
            }
        }
    }

    if (cache == null)
    {
        App.Log("Creating a new cache");
        cache = new RSSCache();
    }

    return cache;
}

```

The following code shows how the application writes data to the storage

```

public void Save()
{
    App.Log("Persisting cache to file");
    using (IsolatedStorageFile file = IsolatedStorageFile.GetUserStoreForApplication())
    {
        using (IsolatedStorageFileStream stream = new
IsolatedStorageFileStream("rssCache.dat", System.IO.FileMode.Create, file))
        {
            DataContractSerializer serializer = new DataContractSerializer(typeof(RSSCache));
            stream.Flush();
            serializer.WriteObject(stream, this);
        }
    }
}

```

### 3.3.3 User Interface

When the application installed in the Windows phone, it will appear in the Windows phone application menu with the name as “Best Sale”. Figure 9 shows the Windows phone application menu after installing the Best Sale application.

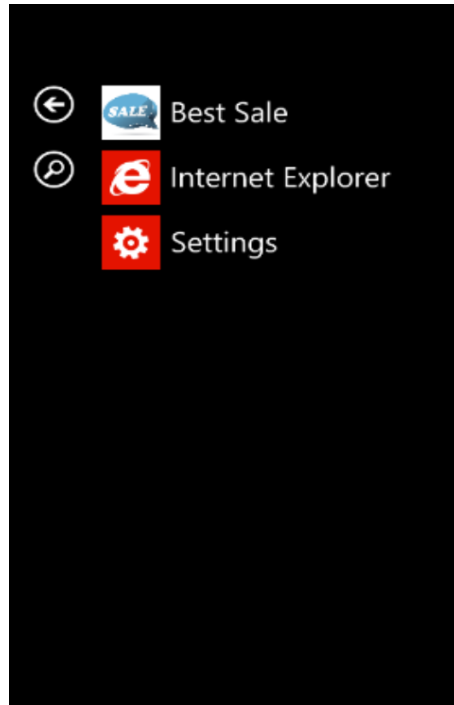


FIGURE 9. Best Sale Application in the Windows Phone Menu

Figure 10 shows a splash screen that will be shown for a few seconds after launching the application from the menu.



FIGURE 10. Best Sale Application Splash Screen Page

Figure 11 shows the first screen of the application showing the list of City names and shop names under each city in a panorama view.



FIGURE 11. Best Sale Application Main Panorama Page

From the panorama page the user can choose another view with a list of products as images. Figure 12 shows a view in which the user has chosen Oulu K-Market. From this page the user can scroll right to see other products.



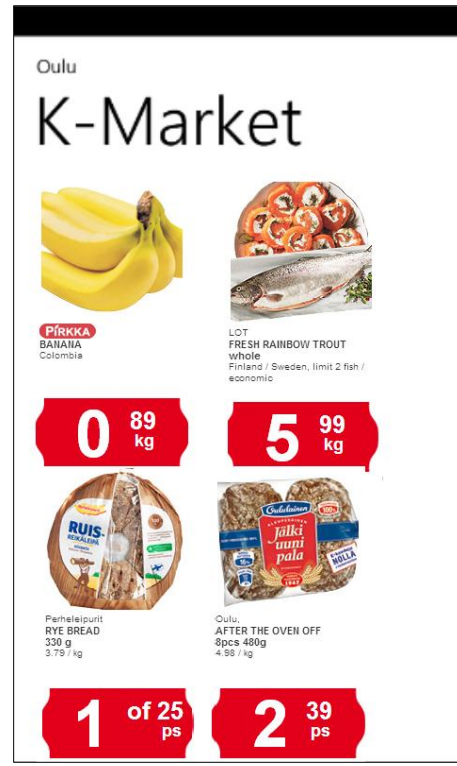


FIGURE 12. Beset Sale Application Store Product View

If the user wants to see more details about the product, he/she can tap on the image which opens another view where the user can see the product description and the dates during which the offer is valid.

### 3.3.4 Testing

The tests were carried out with Best Sale Server. Adding and removing data in the Best Sale Server should reflect in the Best Sale Application on the mobile.

Table 4-9 contain the tests that were done with the prototype application.

TABLE 4. Application Launch Test Case

|                  |   |
|------------------|---|
| Test Id : 1.1    | Test Title<br>Application Launch Test Case.                         |
| Test Description | Install the application in Windows Phone and launch the application |
| Pre Condition    | Application in installed in the device                              |
| Result           | Pass  |

TABLE 5. Application City List Verification Test Case

|                  |  |
|------------------|--|
| Test Id : 1.2    | Test Title<br>Application City List Verification Test Case                         |
| Test Description | Check the city names that are added in the back end server displayed in the device |
| Pre Condition    | Oulu, Tampere , Helsinki city names are added in the back end server               |
| Result           | Pass   |

TABLE 6. Shop Name Verification Test Case

|                  |   |
|------------------|---|
| Test Id : 1.3    | Test Title<br>Shop Name Verification Test Case  |
| Test Description | Check the shop names that are added in the back end server is displayed in the device |
| Pre Condition    | K-Market, S-Market, Verkkokauppa are added under Oulu city in the back end server     |
| Result           | Pass  |

TABLE 7. Product List Verification Test Case

|                  |  |
|------------------|--|
| Test Id : 1.4    | Test Title<br>Check the list of products   |
| Test Description | Check the products added in the back end server is displayed in the device under the shop. |
| Pre Condition    | 10 products are added under Oulu K-Market in the back end server.                          |
| Result           | Pass   |

TABLE 8. Product Details Verification Test Case

|                  |  |
|------------------|--|
| Test Id : 1.5    | Test Title<br>Product Details Verification Test Case.                                    |
| Test Description | Check the products description and offer valid of a product added in the back end server |
| Pre Condition    | A new product is added in the back end server under Oulu K-Market.                       |
| Result           | Pass   |

TABLE 9. Removed Product Verification Test Case

|                  |   |
|------------------|---|
| Test Id : 1.6    | Test Title<br>Removed Product Verification Test Case  |
| Test Description | A shop is removed from the back end server and the application should not display the shop anymore. |
| Pre Condition    | Delete Oulu K-market from the back end server.  |
| Result           | Pass  |

## 4. FUTURE R&D POSSIBILITIES

The prototype version of Best Sale Mobile Application has a limited functionality and it is developed for only one platform. In this chapter the possibilities of new features and porting to different platforms are discussed.

### 4.1 Extending Best Sale App Functionality.

The prototype version of Best Sale Application can be extended to adopt the following new functionalities.

#### Tiles Update:

The Windows Phone platform provides a feature to create a shortcut of the Application as a live tile on the home screen of the Windows Phone. The Best Sale application can run in the background once every hour to get an updated content from the back end server and display the updated notifications into the tile on the Windows Phone home screen.

A code example for setting up a project with tiles [5]

```
<?xml version="1.0" encoding="utf-8"?>
<Deployment
  xmlns="http://schemas.microsoft.com/windowsphone/2009/deployment"
  AppPlatformVersion="7.1">
  <AppExtra xmlns="" AppPlatformVersion="8.0">
    <Extra Name="Tiles" />
  </AppExtra>
  <App ...>
  ...
  </App>
</Deployment>
```

A code example for using new tiles [5]

```
var tileId = ShellTile.ActiveTiles.FirstOrDefault();
if (tileId != null)
{
    var tileData = new FlipTileData();
    tileData.Title = "My app";
    tileData.BackContent = "";
    tileData.BackgroundImage = new Uri("/Images/Icon173x173.png",
```

```

UriKind.Relative);
    tileData.BackBackgroundImage = new
Uri("/Images/Icon173x173_back.png", UriKind.Relative);
    tileData.WideBackContent = "";
    tileData.WideBackgroundImage = new
Uri("/Images/Image346x173.png", UriKind.Relative);
    tileData.WideBackBackgroundImage = new
Uri("/Images/Image346x173_back.png", UriKind.Relative);
    Debug.WriteLine("Activating live tile: " +
Mangopollo.Utills.CanUseLiveTiles);
    tileId.Update(tileData);
}

```

## **New Sale Notification:**

Once the tile update functionality has been implemented, the background best sale application should update the tile data when there is a new sale started in one of the stores listed in the application.

## **Interested Store Notification:**

Best Sale Application should give a freedom to customize the notification for the interesting stores only. Once the user selects the interesting stores, then a notification should be given only for those stores.

## **Share:**

Users might be interested in sharing hot sale deals with their friends and relatives.

Best sale application should provide an option to share the sale product information via Email, SMS and Social Network (Facebook, Twitter).

A code example for sharing via social network

```

ShareLinkTask shareLinkTask = new ShareLinkTask();

```

```
shareLinkTask.Title = "Code Samples";

shareLinkTask.LinkUri = new Uri("http://code.msdn.com/wpapps",
UriKind.Absolute);

shareLinkTask.Message = "Here are some great code samples for
Windows Phone.";

shareLinkTask.Show();
```

## **4.2 Implementation for PC, Android and iOS.**

Best Sale Application should be ported to other widely used mobile platforms such as Android and iOS. Also, it should be available as a browser based version so that it could be accessed from laptops, PCs and tablets.

### **Desktop version:**

The design should not be complicated for a PC version as modern computer programs provide a rich set of browser UI tools and programming scripts. The only limitation could be the notification in browser. This can be solved if a user could provide his email id for interesting stores then a notification can be sent to the user registered email id from the back end server.

### **Android OS:**

Android OS is based on Linux kernel which is primarily designed for touch UI mobile devices. There are many Android based mobile devices released in recent years. Samsung plays a major role in developing Android based mobile devices. Android is also used as an Operating System for Tablets. Google and Samsung have released tablets based on Android OS. The Android OS is owned by Google.

## iOS :

iOS was the leading Mobile OS before Android was introduced in the market. iOS is owned by Apple. iPhone models are one of the most sold devices in the world.

Before starting to develop the Best Sale Application for Android or iOS, the differences should be considered. Table 10 describes the main differences between Android and iOS operating systems.

TABLE 10. Android and iOS differences[6]

| Differences                       | Android                            | iOs  |
|-----------------------------------|------------------------------------|--|
| <b>Widgets</b>                    | Yes                                | No, except in NotificationCenter   |
| <b>Company/Developer</b>          | Google                             | Apple Inc.   |
| <b>OS family</b>                  | Linux                              | OS X, UNIX   |
| <b>Customizability</b>            | A lot. Can change almost anything. | Limited unless jailbroken  |
| <b>Initial release</b>            | September 23, 2008                 | July 29, 2007  |
| <b>Programmed in</b>              | C, C++, java                       | C, C++, Objective-C  |
| <b>Dependent on a PC or a Mac</b> | No                                 | No   |
| <b>Easy media transfer</b>        | depends on model                   | with desktop application   |
| <b>Source model</b>               | Open source                        | Closed, with open source components.   |
| <b>Open source</b>                | Kernel, UI, and some standard apps | The iOS kernel is not open source but is based on the open-source Darwin OS. |
| <b>Call features supported</b>    | Auto-respond                       | Auto-respond, call-back  |

|                              |  |   |
|------------------------------|--|---|
|                              |  | reminder, do not disturb mode   |
| <b>Internet browsing</b>     | Google Chrome (or Android Browser on older versions; other browsers are available)   | Mobile Safari (Other browsers are available)  |
| <b>Available on</b>          | Many phones and <a href="#">tablets</a> , including Kindle Fire(modified android), LG, HTC, Samsung, Sony, Motorola, Nexus, and others.  | iPod Touch, iPhone, iPad, <a href="#">Apple TV</a> (2nd and 3rd generation)                                       |
| <b>Interface</b>             | Touch screen, Smartwatch   | Touch screen  |
| <b>Messaging</b>             | Google Hangouts  | iMessage  |
| <b>Voice commands</b>        | Google Now (on newer versions)   | Siri  |
| <b>Maps</b>                  | Google Maps  | Apple Maps  |
| <b>Video chat</b>            | Google Hangouts  | Facetime  |
| <b>App store</b>             | Google Play – 1,000,000+ Apps. Other app stores like Amazon and Getjar also distribute Android apps. (unconfirmed ".APK's")  | Apple app store – 1,000,000+ Apps   |
| <b>Market share</b>          | 81% of smartphones, 3.7% of tablets in North America (as of Jan'13) and 44.4% of tablets in Japan (as of Jan'13). In the United States in Q1 2013 - 52.3% phones, 47.7% tablets. | 12.9% of smartphones, 87% of tablets in North America (as of Jan'13) and 40.1% of tablets in Japan (as of Jan'13) |
| <b>Available language(s)</b> | 32 Languages   | 34 Languages  |
| <b>Latest stable release</b> | Android 4.4 Kitkat (October, 2013)   | 7.1 (March 10, 2014)  |
| <b>Device manufacturer</b>   | Google, LG, Samsung, HTC, Sony,  | Apple Inc   |



|  |                               |           |
|--|-------------------------------|-----------|
|  | ASUS, Motorola, and many more |           |
| <b>Upcoming releases/Release dates</b> | Unknown                       | Unknown   |
| <b>Working state</b>                   | Current                       | Current   |
| <b>Website</b>                         | android.com                   | apple.com |

### 4.3 Effort Estimations.

A Desktop version of Best Sale Application can be developed by a Web Programmer in 4 weeks including the testing and integration with the backend server. An iOS version of Best Sale Mobile Application can be developed by an iOS developer in 6 weeks. An Android version of Best Sale Mobile Application can be developed by an Android developer in 8 weeks. It takes 2 weeks more than the iOS version because of a Widget implementation and its testing will take more time whereas iOS does not have widget concepts.

## 5. CONCLUSIONS

The Best Sale Application is the first version of reaching a consumer faster with a short-term sale advertisement. It reduces the paper waste and encourages the seller to identify the potential customer interested in the product. This thesis covers the application developed for Windows Phone only.

As a feature development the same application can be developed for other mobile operating systems and a browser version can help consumers who prefer to use their laptop or PC to check the offers.

This application was developed for a thesis purpose and it can be extended to a real world product once there is a potential seller who is ready to offer their product details in an RSS format. This application can be deployed to the customer mobile using a Windows Market place.

## 6. REFERENCES

- 1) Windows Phone Architecture. 2014. Date of retrieval 14 April 2014.  
[http://msdn.microsoft.com/enus/library/windowsphone/develop/ff967556\(v=vs.105\).aspx](http://msdn.microsoft.com/enus/library/windowsphone/develop/ff967556(v=vs.105).aspx)
- 2) Nathan Krasney. 2012. Mobile Software. Date of retrieval 6 February 2012.  
<http://mobilesoftwarenk.blogspot.fi/2012/02/wp-application-architecture-6.html>
- 3) Scripting News, Inc. 2006. OPML structure. Date of retrieval 16 February 2006.  
<http://dev.opml.org/spec2.html>
- 4) K-Citymarket Images for testing. 2014. Date of retrieval 5 January 2014.  
[http://www.k-citymarket.fi/tarjoukset/Ruoka/?utm\\_source=asiakaskirje&utm\\_medium=click&utm\\_campaign=asiakaskirje](http://www.k-citymarket.fi/tarjoukset/Ruoka/?utm_source=asiakaskirje&utm_medium=click&utm_campaign=asiakaskirje)
- 5) Windows Phone 7.8 Tutorail. 2013. Date of retrieval 25 October 2013.  
[http://developer.nokia.com/community/wiki/Windows\\_Phone\\_7.8\\_Tutorial](http://developer.nokia.com/community/wiki/Windows_Phone_7.8_Tutorial)
- 6) Nikhilesh Jasuja, Gauri S, Poonam S, Pooja Sehgal. 2014. Android vs iOS. Date of retrieval 13 May 2014.  
[http://www.diffen.com/difference/Android\\_vs\\_iOS](http://www.diffen.com/difference/Android_vs_iOS)